6Д

УДК 629.3.072.1:004.896

А.Ю. Дорогов, В.Ю. Лесных, И.В. Раков, Г.С. Титов
Санкт-Петербургский государственный электротехнический университет, Россия
vaksa2006@yandex.ru

Алгоритмы оптимального движения мобильных объектов по пересеченной местности и транспортной сети

Исследованы алгоритмы построения оптимальных и квазиоптимальных маршрутов движения мобильных объектов по пересеченной местности и транспортной сети. Рассмотрены алгоритмы для комбинированных вариантов движения. Эффективность предложенных алгоритмов не хуже базового алгоритма Форда-Беллмана и зависит от сложности транспортного графа. Для построения квазиоптимальных решений предложен волновой алгоритм с вычислительной эффективностью, пропорциональной числу узлов транспортного графа.

Введение

Задача нахождения оптимальных путей по уровню транспортных затрат актуальна для ряда технических приложений. В частности, к ним относятся: оценка транспортной доступности для территориально-распределенных систем охраны, планирование оптимальных маршрутов движения робототехнических систем на пересеченной местности, моделирование прокладки маршрутов в тренажерах мобильных систем и компьютерных играх.

Задача планирования оптимального пути в общей постановке формулируется следующим образом. На карте местности необходимо определить маршрут движения от стартового множества точек к множеству конечных точек, обладающий минимальными транспортными затратами. В такой постановке начальные и конечные точки заведомо не известны и определяются в процедуре расчета. В частной постановке возможны следующие варианты задач:

- проложить оптимальный маршрут от множества стартовых точек к заданной конечной точке;
- проложить оптимальный маршрут от множества стартовых точек к заданному множеству конечных точек;
- проложить оптимальный маршрут от заданной стартовой точки к заданной конечной точке;
- построить фронт транспортной доступности с заданным уровнем затрат по отношению к множеству стартовых точек.

Известный базовый алгоритм прокладки оптимальных маршрутов основан на методе динамического планирования Форда-Беллмана на взвешенных графах (1956 – 1958 гг.) [1-3]. В приложении к транспортной задаче на пересеченной местности вершинами графа являются центры элементарных участков карты, а дуги соответствуют переходам между центрами смежных участков. Для транспортной сети вершинами графа являются узлы транспортной сети, а дуги соответствуют переходам между узлами. Множество алгоритмов, предложенных в последующие годы (алгоритмы Дейкстры, Калаба, А-звезда и др.), в основном являются вариа-

циями базового алгоритма для частных постановок, благодаря чему достигается более высокая вычислительная эффективность данных алгоритмов по сравнению с базовым алгоритмом.

В настоящей работе рассматривается алгоритм оптимального планирования маршрутов в общей постановке. В отличие от алгоритма Форда-Беллмана маршрут строится в виде последовательно улучшаемых решений, причем на первом этапе для прокладки маршрута используется волновой алгоритм с вычислительной сложностью, пропорциональной числу узлов. На последующих этапах маршрут уточняется за счет последовательного ослабления дуг. Алгоритм завершается при стабилизации накопленных затрат для узлов графа. В целом вычислительная сложность алгоритма не превышает вычислительной сложности алгоритма Форда-Беллмана, однако возможность построения приближенных решений позволяет существенно сократить число вычислительных операций в тех случаях, когда в конкретной задаче требования к точности решений можно понизить.

В разделе 1 представлен алгоритм выбора маршрута при движении по пересеченной местности. В разделе 2 представлен модифицированный вариант для использования в транспортной сети. В разделе 3 обсуждается построение оптимальных и квазиоптимальных маршрутов при комбинированном движении по пересеченной местности и транспортной сети.

1. Алгоритм выбора оптимального маршрута на пересеченной местности

1.1. Модель местности

Карта местности представлена в виде матрицы $Z = z(y, x)$, где каждый элемент определяет максимальные затраты на преодоление участка с координатами (y, x). Участок местности, соответствующий элементу матрицы, считается квадратом со стороной d, правильно ориентированным вдоль координатных осей. Под координатами участка понимаются матричные индексы: y – номер строки матрицы, x – номер столбца. Максимальные затраты соответствуют пересечению участка по диагонали. Полагается, что затраты при пересечении вдоль любой координатной оси в $\sqrt{2}$ раз меньше максимальных.

При переходе с данного участка на смежный участок будем считать, что переход осуществляется из центра одного участка в центр другого. Затраты на переход равны среднему значению затрат по двум смежным участкам:

$$z_{12} = (z_1 + z_2)/2 .$$

Затраты на маршруте определяются суммой затрат по всем переходам между точками маршрута:

$$Q = \sum_{i=1}^{N-1} z_{i,i+1} ,$$

где N – число точек маршрута. При построении оптимального маршрута на каждом шаге необходимо выбирать направление дальнейшего движения. Результат выбора называется шаговым управлением и обозначается c_k. Управление всей операцией состоит из совокупности шаговых управлений:

$$C = (c_1, c_2, \ldots c_{m-1}).$$

Маршруты задаются координатами элементарных участков карты, поэтому шаговые управления и накопленные затраты для множества маршрутов представляются в виде матриц $C = c(y, x)$ и $Q = q(y, x)$, а шаговое перемещение реализуется в

пределах маски размером 3 × 3. Это позволяет интерпретировать матрицу накопленных затрат в виде полутонового изображения и использовать методы рекурсивной пространственной фильтрации для нахождения оптимальных маршрутов. Начальное заполнение матрицы накопленных затрат реализуется волновым алгоритмом с числом вычислительных операций, пропорциональным числу элементов матричной карты. Начальное заполнение дает первую оценку оптимальных маршрутов, которая в ряде случаев уже является удовлетворительной.

1.2. Фильтрующий алгоритм

Принцип динамического планирования Беллмана можно выразить фразой: «любая часть оптимальной траектории оптимальна». В частности, для любой промежуточной точки оптимального маршрута минимальна сумма затрат от стартового множества до данной точки. На основе принципа Беллмана можно предложить следующий вариант алгоритма решения общей задачи оптимального планирования пути.

Начиная от точек стартового множества, строится веер оптимальных траекторий до тех пор, пока некоторая оптимальная траектория не достигнет какой-либо конечной точки. На каждом шаге алгоритма запоминается оптимальное направление перехода в текущую точку (шаговое управление на предшествующем шаге), что позволяет при достижении конечной точки восстановить весь оптимальный маршрут.

В предлагаемом алгоритме матрица накопленных затрат Q и матрица направлений C интерпретируются как двумерные изображения. Значение каждого элемента матрицы соответствует уровню яркости изображения. Алгоритм построения веера оптимальных траекторий реализуется как многоступенчатый пространственный параметрический ранговый фильтр минимума с маской размером 3×3 (рис. 1).

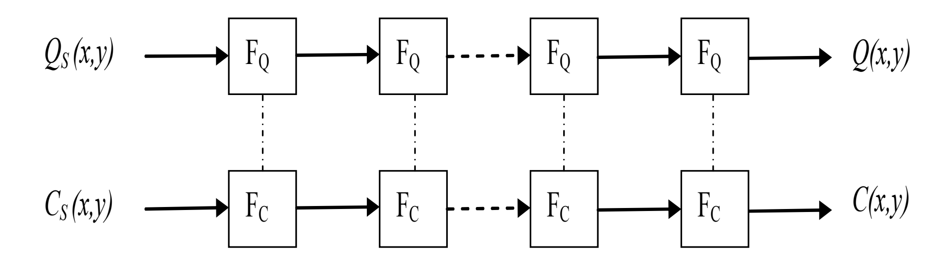


Рисунок 1 – Многоступенчатый фильтр оптимальных затрат

$(y-1, x-1)$	$(y-1, x)$	$(y-1, x+1)$
$(y, x-1)$	(y, x)	$(y, x+1)$
$(y+1, x-1)$	$(y+1, x)$	$(y+1, x+1)$

Рисунок 2 – Маска пространственного фильтра

Фильтрация производится над матрицами накопленных затрат $Q = q(y, x)$ и матрицей шаговых управлений $C = c(y, x)$. Матрица $Z = z(y, x)$ определяет параметр фильтра для каждой точки (элемента) матриц Q и C. Веер траекторий строится от

множества стартовых точек и распространяется по всем направлениям карты. Эффект пространственной фильтрации матриц эквивалентен ослаблению дуг в алгоритме Форда-Беллмана.

Размещение фильтрующей маски в точке с координатами (y, x) показано на рис. 2. Число ступеней фильтра заведомо не определено, каждая ступень улучшает предыдущее решение. Матрица затрат Z содержит относительные затраты, приведенные к диапазону $[0,1]$ (0 – минимальные затраты, 1 – максимальные затраты в пределах зоны). Непреодолимые области имеют значение ∞. Матрица может иметь необрабатываемые области, которые помечены символом NaN (символика Матлаб). Значение элемента матрицы $z(y, x)$ используется как параметр пространственного фильтра в точке (y, x). Матрица накопленных затрат $Q = q(y, x)$ имеет размер матрицы Z и содержит следующие области:

– необрабатываемая область – точки помечены символом NaN. При выполнении пространственной фильтрации эти точки пропускаются и не участвуют в обработке;
– стартовые точки (в пределе это может быть одна точка) определяют границу, от которой распространяется волновой процесс накопления затрат. В начальном состоянии стартовые точки заполняются соответствующими значениями из матрицы затрат Z;
– точки, до которых волновой фронт еще не дошел – имеют значение ∞. В процессе выполнения алгоритма значение этих точек изменяется и в конце первой ступени алгоритма содержит оценку минимальных затрат на движение от множества стартовых точек к данной точке.

Матрица шаговых управлений $C = c(y, x)$ имеет размеры матрицы Z. Каждая ячейка матрицы содержит структуру из двух координат. В процессе выполнения алгоритма в ячейках матрицы C устанавливаются значения координат, которые указывают на соседнюю предшествующую точку оптимального маршрута. Можно сэкономить память, если вместо пары координат в ячейке $c(y, x)$ сохранять кодовый номер направления на соседнюю точку. Таких направлений всего 8 (рис. 3).

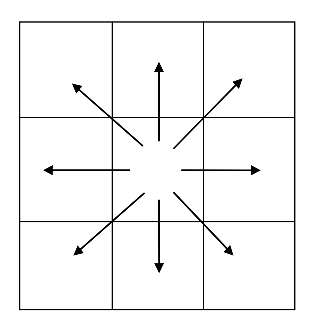


Рисунок 3 – Возможные варианты шаговых управлений

При пространственной фильтрации в классическом варианте маска последовательно перемещается вдоль строк и столбцов изображения, формируя новые значения яркости. Однако в данном случае на первой ступени фильтрации такая реализация оказывается неэффективной, поскольку большинство точек матрицы Q (имеющих бесконечные значения) не граничат со стартовыми точками и поэтому не изменяют свое значение, но, тем не менее, будут просматриваться сканирующей маской.

Поэтому на первой ступени фильтрации маска перемещается только вдоль границы, отделяющей настоящее и прошлое движение от будущего. Вначале граница

охватывает только стартовые точки, а затем, по мере построения веера оптимальных траекторий, она распространяется в виде волны по всей плоскости матрицы Q.

При движении маски вдоль границы только часть точек, покрываемых маской, может быть использована для вычисления накопленных затрат. Это означает, что в процессе вычисления шаговых управлений будут пропущены некоторые направления из-за того, что будущее движение нам пока не известно. После первой ступени фильтрации будут получены оценки накопленных затрат для всех точек матрицы Q (оценки первого приближения). При этом в матрице направлений $C(y,x)$ все элементы будут содержать направления маршрута первого приближения.

Последующие ступени фильтрации позволяют уточнить первое приближение. Поскольку после первой ступени матрицы Q и C полностью определены, то на последующих ступенях фильтрации сканирование маской выполняется последовательно по строкам и столбцам, при этом фильтр маски использует все восемь возможных шаговых управлений (рис. 3).

1.3. Реализация алгоритма

Ступень 1. На первой ступени фильтрации алгоритм распространяется от стартовых точек в виде волны. Фронт волны строится следующим образом. Множество стартовых точек рассматривается как начальный фронт волны. Маска позиционируется в стартовую точку карты и в матрицах затрат Z, Q просматриваются все точки, покрываемые маской. Точки, для которых значения в матрице Z равны ∞, а значения в матрице Q конечны, переносятся в список граничных точек. Просматриваются все стартовые точки и строится общий список. Далее список редактируется, из него удалятся повторяющиеся точки (редактирование может производиться и при смене позиции маски). Логические условия построения нового фронта имеют вид:

$$(y,x) \in Front, \quad \text{если} \quad \left\{ q(y,x) = \infty \quad \& \quad z(y,x) \neq \infty, \neq NaN \quad \& \quad q\left(y^*, x^*\right) \neq \infty \right\} \quad (1)$$

где координаты y и x принадлежат полю движущейся маски, y^*, x^* – координаты точек текущего фронта волны. Фильтрация выполняется для всего списка граничных точек. С этой целью маска позиционируется центром в граничную точку (y,x) и выполняется вычисление нового значения накопленных затрат по следующему правилу:

$$q(y,x) = \min_{i,j}\left(q(y+j, x+i) + z_{ji}(y,x)\right), \quad (2)$$

где $j,i = 0, \pm 1$ – индексы маски с ненулевыми значениями элементов. Полагается, что $z_{00}(y,x) = 0$. В центральных областях карты матрица фильтрующей маски имеет вид:

$$Maska = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Матрица маски наполняется нулями при пересечении границы карты. Значение соответствующего элемента матрицы шаговых управлений устанавливается по позиции точки минимума в выражении (2):

$$c(y,x) = \arg\min_{i,j}\left(q(y+j, x+i) + z_{ji}(y,x)\right). \quad (3)$$

Полученный фронт используется для построения нового списка граничных точек. Процедура волновой фильтрации повторяется для нового списка граничных точек. Итерационный процесс заканчивается, когда новый фронт не содержит ни одной точки. После выполнения первой ступени все точки матрицы накопленных затрат, которые были равны ∞, получают конечное положительное значение.

Ступень 2 и последующие. Последовательно просматриваются все точки карты, для которых $q(y,x) \neq NaN$ и выполняется коррекция значений матриц Q и C, следуя формулам (2) и (3). Фильтрация заканчивается, когда матрицы Q двух смежных ступеней совпадают. Дополнительный эффект ускорения вычислений достигается за счет рекурсивного принципа фильтрации.

Построение оптимального маршрута. При построении маршрута возможны следующие варианты:

1. Стартовых точек несколько. Необходимо проложить маршрут от ближайшей стартовой точки к заданной конечной точке с координатами (y_k, x_k). В этом случае точки маршрута восстанавливаются по цепочке шаговых управлений:

$$x_{k-1} = c(y_k, x_k).x, \quad y_{k-1} = c(y_k, x_k).y.$$

2. Конечная точка не задана, но определено множество, которому она принадлежит. В этом случае на заданном множестве необходимо найти точку, в которой $q(y,x) \to \min$ и далее повторить алгоритм пункта 1.

3. Если заданы начальная и конечная точки, то алгоритм необходимо выполнить для начального фронта, состоящего из одной стартовой точки.

Построение фронта транспортной доступности. В матрице накопленных затрат ищутся точки, удовлетворяющие условию: $q(y,x) \approx L$, где L – уровень затрат на линии фронта. Степень приближения может быть задана, например, в процентах от уровня L.

2. Алгоритм оптимального движения в транспортной сети

2.1. Модель транспортной сети

Транспортная сеть (рис. 4) представляется топологической моделью в виде взвешенного графа G. Ребра графа соответствуют однородным сегментам транспортной сети, а узлы – точкам ветвления или иным точкам интереса (стартовым точкам, конечным точкам, точкам сети, ближайшим к заданному объекту и т.д.). Каждому ребру графа поставлено в соответствие значение временных затрат на преодоление сегмента. Сегментные затраты определяются геометрической длиной сегмента и средней скоростью движения объекта вдоль сегмента.

Постановка задачи по прокладке маршрута в транспортной сети не отличается от задачи маршрутизации по пересеченной местности.

2.2. Описание алгоритма

Обозначим через S множество стартовых точек и через E множество конечных точек. Для каждого узла сети введем понятие накопленных затрат. Накопленные затраты определяются минимальными суммарными затратами на перемещение из ближайшей стартовой точки в данную точку.

Начиная от точек стартового множества, строится веер оптимальных траекторий до тех пор, пока некоторая оптимальная траектория не достигнет какой-либо конечной точки. В каждой узловой точке производится вычисление минимальных

накопленных затрат по отношению к ближайшей стартовой точке. Кроме того запоминается сегмент оптимального маршрута (оптимальное шаговое управление на предшествующем шаге), что позволяет при достижении конечной точки восстановить весь оптимальный маршрут.

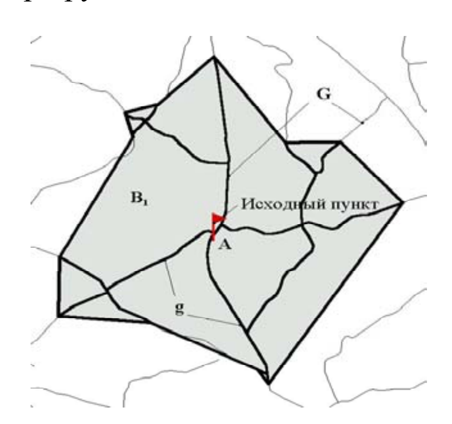


Рисунок 4 – Транспортная сеть

2.3. Реализация алгоритма

Взвешенный граф транспортной сети задается квадратной матрицей затрат $Z = z(i, j)$. Размер матрицы определяется числом вершин в графе. Каждый элемент матрицы представляет собой затраты на перемещение между смежными вершинами. В общем случае матрица может быть не симметрична ($z(i, j) \neq z(j,i)$). Если вершины не являются смежными, то значение элемента матрицы равно ∞. Кроме того элемент матрицы может иметь 0, если $i = j$ или значение NaN (символика Матлаб) – такой сегмент не участвует в построении оптимального маршрута.

Накопленные затраты сохраняются в массиве Q. Каждый элемент массива соответствует вершине графа. Первоначально всем элементам массива Q присваивается значение ∞. Элементу массива Q присваивается значение NaN, если все сегменты окружения для соответствующей вершины графа имеют значение NaN.

Ступень 1. На первой ступени алгоритма накопление затрат происходит в виде волны от стартовых точек. Фронт волны строится следующим образом. Множество стартовых точек рассматривается как начальный фронт волны. Алгоритм фокусируется в стартовой точке, в матрице весов Z и массиве накопленных затрат Q. Просматриваются все сегменты и узлы ближайшего окружения. Точки, для которых значения в массиве Q равны ∞, и при этом соответствующие значения в матрице сегментных затрат Z конечны, переносятся в список граничных точек. Просматриваются все стартовые точки и строится общий список. Далее список редактируется, из него удалятся повторяющиеся точки. Логическая формула построения нового фронта имеет вид:

$$i \in Front \text{ если} \{q(i) = \infty \ \& \ z(i, j) \neq \infty, NaN\}. \quad (4)$$

Фильтрация первой ступени[1]. Алгоритм фокусируется в граничную точку i и выполняется вычисление нового значения накопленных затрат, следуя правилу:

$$q(i) = \min_{j \in q^{-1}(i)} \left(q(j) + z_{ji}\right). \quad (5)$$

[1] Применительно к графам используется терминология «ослабление дуг».

Значение соответствующего элемента массива шаговых управлений устанавливается по позиции точки минимума в выражении (5):

$$c(i) = \arg \min_{j \in q^{-1}(i)} \left(q(j) + z_{ji} \right). \qquad (6)$$

Когда все точки текущего фронта пройдены, строится новый фронт по вышерассмотренному алгоритму. Итерационный процесс первой ступени заканчивается, когда новый фронт не содержит ни одной точки. После выполнения первой ступени все точки массива накопленных затрат Q, которые были равны ∞, получают конечное положительное значение.

Ступень 2 и последующие. Последовательно просматриваются все точки зоны, для которых $q(i) \neq NaN$ и выполняется коррекция значений, массивов Q и C, следуя выражениям (5) и (6). Фильтрация заканчивается либо когда массивы двух смежных ступеней совпадают, либо после выполнения заданного числа ступеней фильтрации.

Построение оптимального маршрута. При построении маршрута возможны следующие варианты:

1. Стартовых точек несколько. Необходимо проложить маршрут от ближайшей стартовой точки s_{i_1} к заданной конечной точке e_{i_k}. В этом случае узловые точки маршрута восстанавливаются по цепочке шаговых управлений: $i_{k-1} = c(i_k)$.
2. Конечная узловая точка не задана, но определено множество, которому она принадлежит. В этом случае на заданном множестве необходимо найти узловую точку, в которой $q(i_k) \to \min$ и далее повторить алгоритм пункта 1.
3. Если заданы начальная и конечная узловые точки, то алгоритм формирования массива накопленных затрат необходимо выполнить для начального фронта, состоящего из одной стартовой точки.

Построение фронта транспортной доступности.

Фронт транспортной доступности представляет собой линию (изохрону), в каждую точку которой можно попасть из стартовой точки примерно при одном и том же уровне затрат L. В массиве накопленных затрат Q ищутся точки, удовлетворяющие условию: $q(i) \approx L$. Степень приближения должна быть задана (например, в процентах от уровня L). Узловым точкам изохроны соответствуют координаты (x_i, y_i). По этим координатам можно приближенно построить фронт транспортной доступности, соединив их отрезками прямых линий.

3. Алгоритм прокладки оптимального маршрута при комбинированном движении

Комбинированное движение состоит из этапа движения по транспортной сети с использованием транспортных средств и этапа движения по пересеченной местности.

3.1. Построение первого приближения

1. С помощью волнового алгоритма определяются достижимые затраты на перемещение из стартового узла во все узлы транспортной сети.

2. Узлы транспортной сети рассматриваются как стартовые точки для оптимального алгоритма движения по пересеченной местности, причем для стартовых точек задается начальный уровень затрат, полученный при расчете движения по транспортной сети.

3. С помощью волнового алгоритма рассчитывается оптимальный маршрут движения по пересеченной местности от множества стартовых точек.

4. Выделяется оптимальная стартовая точка (узел транспортной сети, принадлежащий оптимальному маршруту).

Точность первого приближения определяется максимальным расстоянием между оптимальной стартовой точкой и смежными с ней узлами.

3.2. Уточнение решения

1. Сегменты транспортной сети, инцидентные оптимальной стартовой точке, детализируются дополнительным числом узлов.
2. С помощью волнового алгоритма определяются достижимые затраты на перемещение из стартового узла к вновь образованным дополнительным узлам транспортной сети. С помощью этапов фильтрации (ослабления дуг) уточняется решение на транспортной сети.
3. Множество дополнительных узлов сети вместе с оптимальной стартовой точкой рассматриваетсяё как стартовые точки для оптимального алгоритма движения по пересеченной местности, причем для стартовых точек задается уровень затрат, полученный при расчете движения по транспортной сети.
4. С помощью волнового алгоритма рассчитывается оптимальный маршрут движения по пересеченной местности от множества стартовых точек.
5. Выделяется оптимальная стартовая точка.
6. При необходимости построения точного маршрута выполняются этапы фильтрации в алгоритме движения по пересеченной местности.

Заключение

Представленные алгоритмы покрывают практически значимые варианты прокладки оптимальных маршрутов. Быстродействие алгоритмов зависит от требуемой точности построения маршрута. Для построения квазиоптимальных решений достаточно ограничиться волновым алгоритмом. В этом случае вычислительные затраты минимальны и пропорциональны числу узлов транспортного графа. Дополнительные этапы фильтрации (ослабления дуг) обеспечивают улучшение решений в пределе до оптимального, при этом вычислительная эффективность интегрального алгоритма будет не хуже базового алгоритма Форда-Беллмана. Число дополнительных этапов фильтрации зависит от сложности транспортного графа. Поэтому для конкретной задачи использование предложенных алгоритмов может оказаться значительно более эффективным, чем использование базового алгоритма.

Литература

1. Кофман А. Введение в прикладную комбинаторику. – М.: Наука, 1975. – 479 с.
2. Асанов М.О., Баранский В.А., Расин В.В. Дискретная математика: графы, матроиды, алгоритмы. – Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001. – 288 с.
3. Кормен Томас Х., Лейзерсон Чарльз И., Ривест Рональд Л., Штайн Клиффорд. Алгоритмы: построение и анализ: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 1296 с.

О.Ю. Дорогов, В.Ю. Лєсних, І.В. Раков, Г.С. Тітов

Алгоритми оптимального руху мобільних об’єктів по пересіченій місцевості і транспортній мережі
Досліджено алгоритми побудови оптимальних і квазіоптимальних маршрутів руху мобільних об’єктів по пересіченій місцевості і транспортній мережі. Розглянуто алгоритми для комбінованих варіантів руху. Ефективність запропонованих алгоритмів не гірше базового алгоритму Форда-Беллмана і залежить від складності транспортного графа. Для побудови квазіоптимальних рішень запропоновано хвильовий алгоритм з обчислювальною ефективністю, пропорційною числу вузлів транспортного графа.

Статья поступила в редакцию 30.07.2008.